Screencast: 11-ownership-and-permissions.webm or 11-ownership-and-permissions.mp4
Immutable bit: 11a-immutable-bit.webm or 11a-immutable-bit.mp4

References:
UaLSAH REFERENCE - Chapter 6, The File System
TLCL - Chapter 9, Permissions

We've talked about user accounts and looked at the contents of the /etc/passwd and /etc/shadow files. For a more complete picture it is time to visit the /etc/group file. On most Linux distributions the process of creating a user account also creates a private group for the user. Take for example the user dowdle:

Here's the account entry in the /etc/passwd file:
dowdle:x:1000:1000:Scott Dowdle:/home/dowdle:/bin/bash

Here's the account entry in the /etc/group file:
dowdle:x:1000:

Here's an example of a group with multiple users:
wheel:x:10:root,dowdle

This access control method, primarily controlled by the filesystem, is called Discretionary Access Control (DAC). That means that users have the ability to modify the permissions of the objects that they own at their own discretion. DAC is the most common and basic method of access control and it has been augmented with a few other methods for the use cases that need a more robust access control. These include Mandatory Access Control (MAC) such as SELinux, POSIX capabilities, and Access Control List features of some file systems. For an overview of various access control methods, read sections 4.1 and 4.2 in chapter 4.

With a good grasp of DAC, UID and GID it is time to examine file and directory ownership and permissions.

Related commands

chmod - Change file access permissions
chown - Change file owner and group
chgrp - Change group ownership
lsattr / chattr - Change file attributes (immutable bit) [see related screencast]

Ownership

user (whoami, /etc/passwd)
group (id, /etc/group)

Permissions

Symbolic representation
u - Permissions granted to the user who owns the file
g - Permissions granted to users who are members of the file’s group
o - Other, neither u nor g
a - u, g, and o
r - Read
w - Write
x - Execute (or access for directories)

X - Execute only if the file is a directory or already has execute permission for some user
s - Set user or group ID on execution
t - sTicky

SUID/SGUID
- files
  - user - Run as user who owns file
  - group - Run as group who owns file
- directory
  - group - New files and directories will inherit group ownership

sticky
- files - Not used by Linux
- directories - When the sticky bit is set on a directory, files in that directory may be unlinked or renamed only by root or their owner. Without the sticky bit, anyone able to write to the directory can delete or rename files. The sticky bit is commonly found on directories, such as /tmp, that are world-writable.

Octal representation

A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1.
Any omitted digits are assumed to be leading zeros.
nnnn
The first digit selects the set user ID (4) and set group ID (2) and sticky (1) attributes.
4 - read
2 - write
1 - execute
- 0755 -rwxr-xr-x
- 0644 -rw-r--r--
- 0400 -r--------

Examples

Use "ls -l" to see information about a file and "ls -ld" to see information about a directory.

drwx------ 2 dowdle dowdle 4096 Jan 31 01:04 /home/dowdle

-rwsr-xr-x 1 root root 19904 Jan 6 2007 /usr/bin/passwd

drwxrwsr-x 6 root project 4.0K Jan 23 22:06 project_compiler

You will notice that there are:
- 10 placeholders
  - The first is for the type of object
    - d = directory
    - - = file
    - b = block device
    - c = character device
    - l = symbolic link
    - s = socket
    - p = named pipe
  - The following 9 are broken up into
    - user / owner

read
write
execute
group
read
write
execute
other (everyone else)
read
write
execute

Device files in /dev are a little different and have major and minor driver attributes

Another thing to take into account is that the execution bit is overloaded to include setuid, setgid, and sticky.

link count
owner
group owner
size
modification date
object name

umask - File creation mask

Built-in bash command used to set the default create mode of files and directories

Example:

```
[root@kvm-63 ~]# umask
0022
[root@kvm-63 ~]# touch file
[root@kvm-63 ~]# ls -l file
-rw-r--r-- 1 root root 0 Feb 2 20:46 file
[root@kvm-63 ~]# rm file
rm: remove regular empty file `file'? y
[root@kvm-63 ~]# umask 0577
[root@kvm-63 ~]# touch file
[root@kvm-63 ~]# ls -l file
--w------- 1 root root 0 Feb 2 20:47 file
```